UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/605,448	09/30/2003	Kevin Scott Beyer	SVL920030088US1	2447

45727 7590 09/16/2008

IP AUTHORITY, LLC
RAMRAJ SOUNDARARAJAN
4821A Eisenhower Ave
Alexandria, VA 22304

EXAMINER	
MYINT, DENNIS Y	

ART UNIT	PAPER NUMBER
2162	

NOTIFICATION DATE	DELIVERY MODE
09/16/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

RAM@IP-AUTHORITY.COM
brandi@ip-authority.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/605,448
Filing Date: July 7, 2008
Appellant(s): Kevin Scott Beyer et al.,

Ramraj Soundararajan

For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed on July 7, 2008, appealing from the Office action mailed January 23, 2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

A statement identifying the related appeals and interferences, which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief.

(3) Status of Claims

The statement of the status of the claims contained in the brief is correct.

Claims 23-44 are pending in this application and were finally rejected in the Final Office Action mailed on January 23, 2008.

Claims 23-44 are the subject of this appeal.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejections to be Reviewed on Appeal

The Appellant's statement of grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,889,226 B2	O'Neil et al.,	3-2005
2004/0068500 A1	Rizzo et al.,	4-2004

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims.

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

3. Claim 23-44 are rejected under 35 U.S.C. 103(a) as being unpatentable over O'Neil et al., (hereinafter "O'Neil") (U.S. Patent Number 6889226) in view of Rizzo et al., (hereinafter "Rizzo") (U.S. Patent Application Publication Number 2004/0068500A1).

As per claim 23, O'Neil is directed to a robust computer-based method for updating a computer-stored hierarchical structure of nodes via a node identification technique, said update retaining properties and parent/child relationships of said hierarchical structure without renumbering existing node ID values associated with said hierarchical structure (Abstract, i.e., *The hierarchically organized data is represented as a tree, and each node in the tree is assigned a position identifier that represents both depth level of the node within the hierarchy, and its ancestor/descendant relationship to other nodes*) and teaches the limitations:

(a) "receiving an instruction to insert a new node at an insertion point in said computer-stored hierarchical structure" (O'Neil , Figure 6; Figure 3; Figure 5: Column 8

Lines 36-40, i.e., *Figure 5 and 6 show how data can be inserted (or careted) into a hierarchical data structure, while still maintaining the valuable properties of the position identifier numbering scheme described above*);

(b) “identifying one of, or a combination of the following: a left node ID value closest to the left of said insertion point or a closest right node ID value closest to the right of said insertion point” (O’Neil , Column 8 Lines 58-62, i.e., *In this example, nodes 602 and 604 are assigned position numbers “1.2.1” and “1.2.3”, respectively, now becoming sibling nodes to the right of 504 and to the left of 506*);

(c) “calculating a new ID value based upon node ID value(s) identified in (b), said calculated value greater than ID values of nodes to the left of said insertion point and less than ID values of nodes to the right of said insertion point” (O’Neil , Column 8 Lines 58-62, i.e., *In this example, nodes 602 and 604 are assigned position numbers “1.2.1” and “1.2.3”, respectively, now becoming sibling nodes to the right of 504 and to the left of 506*); and

(d) “(encoding) said calculated new ID value and updating said computer storage storing said nodes of said hierarchical structure with said (encoded) value” (O’Neil , Column 8 Lines 58-62, i.e., *In this example, nodes 602 and 604 are assigned position numbers “1.2.1” and “1.2.3”, respectively, now becoming sibling nodes to the right of 504 and to the left of 506*), “wherein order, node ID values, and relationships between parent, child, and siblings in said hierarchical structure of nodes stored in said storage remain unchanged with said insertion of new node” (O’Neil , Column 8 Lines 36-40, i.e.,

Figure 5 and 6 show how data can be inserted (or careted) into a hierarchical data structure, while still maintaining the valuable properties of the position identifier numbering scheme described above).

O'Neil does not explicitly teach the limitation: "encoding" and "said new ID value based upon a low/high key value, said high key value representing a highest encodable value and said low key value representing lowest encodable value".

On the other hand, Rizzo teaches the limitation:

"encoding" (Rizzo, Paragraph 0056, i.e., *the key field to be represented with B bits in two's-complement system*) and "said new ID value based upon a low/high key value, said high key value representing a highest encodable value and said low key value representing lowest encodable value" (Rizzo, Paragraph 0056, i.e., *It is necessary to represent two symbolic* ***values in the key field*** *data range--specifically a representation for positive infinity (+INF) and negative infinity (-INF) is required. It is assumed the key field to be represented with B bits in two's-complement system and to interpret* ***the greatest positive number*** *(2^{B-1}-1) as +INF* ***and the smallest negative number*** *(-2^{B-1}) as -INF. The key value has therefore a range of [$-2^{B-1}+1$, $2^{B-1}-2$], boundary included*).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify the method of O'Neil to add the features of encoding key values into binary numbering system and employing a range of keys/ID's which fall between highest encodable value and lowest encodable value, as taught by Rizzo, to

the method of O'Neil so that, in the resultant method, new ID value would be based upon a low/high key value, said high key value representing positive infinity and said low key value representing negative infinity. One would have been motivated to do so in order to have a computer interpret positive infinity as the greatest positive number said computer could process and interpret negative infinity as the smallest number said computer could process (Rizzo, Paragraph 0056) and in order to disable further processing when a data structure is empty or to signal when the data structure is full and overflowing (Rizzo, Paragraph 0055).

As per claim 24, O'Neil in view of Rizzo teaches the limitation:

"wherein said new ID value is calculated via any of the following steps: concatenating said left node ID value with one or more high key values and a positive value, decreasing last digit of said right node ID value, increasing last digit of left node ID value, decreasing last digit of said right node ID value and concatenating a positive value, or concatenating said left node ID value with one or more zeros and a positive value" (O'Neil , Column 9 Lines 10-13, i.e., *If node 610 later needs to be inserted in between nodes 608 and 504, the new node 610 will be numbered "1.0.1 (i.e., "0" is the even number between 1 and –1).*)

As per claim 25, O'Neil in view of Rizzo teaches the limitation:

“wherein a digit in said calculated ID value has a negative value” (O’Neil , Column 9 Lines 7-10, i.e., *although insertions to the left of a group of siblings may require a negative odd number – e.g., node 608, which is inserted to the left of the node having position number “1.1”, has position number “1.-1”*).

As per claim 26, O’Neil in view of Rizzo teaches the limitation:

“wherein said encoding is binary encoding" (Rizzo, Paragraph 0056, i.e., *It is necessary to represent two symbolic values in the key field data range--specifically a representation for positive infinity (+INF) and negative infinity (-INF) is required. It is assumed the key field to be represented with **B bits in two's-complement system** and to interpret the greatest positive number (2^{B-1}-1) as +INF and the smallest negative number (-2^{B-1}) as -INF. The key value has therefore a range of [-2^{B-1}+1, 2^{B-1}-2], boundary included*)” and said highest encodable value is 1111 and said lowest value is 0000” (Rizzo, Paragraph 0056, i.e., *It is necessary to represent two symbolic values in the key field data range--specifically a representation for positive infinity (+INF) and negative infinity (-INF) is required. It is assumed the key field to be represented with **B bits in two's-complement system** and to interpret the greatest positive number (2^{B-1}-1) as +INF and the smallest negative number (-2^{B-1}) as -INF. The key value has therefore a range of [-2^{B-1}+1, 2^{B-1}-2], boundary included*). Note that Rizzo teaches a range of key values which are encoded in binary numbering system (i.e., B bits in two’s-complement system). As such, a person of ordinary skill in the art could set the

highest/lowest encodable value in binary numbers to any value, including 1111 and 0000.

As per claim 27, O'Neil in view of Rizzo teaches the limitation:

"wherein said ID values are encoded and are byte comparable" (O'Neil, Column 10 Lines 30-50, i.e., *The following table shows an exemplary set of Li values, and the prefix-property-obedient bit sequences that represent them*).

As per claim 28, O'Neil in view of Rizzo teaches the limitation:

"wherein said nodes are associated with a mark-up language based document" (O'Neil , Column 2 Lines 53-62, i.e., *Extensible Markup Language (XML);* Column 5 Line 38 through Column 6 Line 28, i.e., *Hierarchy structure of data 200*, and *By convention in XML, levels of organization are delimited by*).

As per claim 29, O'Neil in view of Rizzo teaches the limitation:

"wherein said mark-up based language is XML" (O'Neil , Column 2 Lines 53-62, i.e., *Extensible Markup Language (XML)*).

As per claim 30, O'Neil in view of Rizzo teaches the limitation:

"wherein said method is implemented in conjunction with a relational database" (O'Neil , Column 2 Lines 55-60, i.e., *The present invention provides a technique for storing such hierarchical data in a non-hierarchical data structure such as relation, which still maintaining information about the hierarchical structure of the data*).

Claims 31 is essentially the same as claim 23 except that it set forth the claimed invention as an article of manufacture comprising a computer usable medium having readable program code rather than a method and rejected for the same reasons as applied hereinabove.

Claims 32 is essentially the same as claim 24 except that it set forth the claimed invention as an article of manufacture comprising a computer usable medium having readable program code rather than a method and rejected for the same reasons as applied hereinabove.

Claims 33 is essentially the same as claim 27 except that it set forth the claimed invention as an article of manufacture comprising a computer usable medium having readable program code rather than a method and rejected for the same reasons as applied hereinabove.

Claims 34 is essentially the same as claim 28 except that it set forth the claimed invention as an article of manufacture comprising a computer usable medium having readable program code rather than a method and rejected for the same reasons as applied hereinabove.

Claims 35 is essentially the same as claim 29 except that it set forth the claimed invention as an article of manufacture comprising a computer usable medium having readable program code rather than a method and rejected for the same reasons as applied hereinabove.

Claims 36 is essentially the same as claim 30 except that it set forth the claimed invention as an article of manufacture comprising a computer usable medium having readable program code rather than a method and rejected for the same reasons as applied hereinabove.

Claims 37 is essentially the same as claim 26 except that it set forth the claimed invention as an article of manufacture comprising a computer usable medium having readable program code rather than a method and rejected for the same reasons as applied hereinabove.

Claims 38 is essentially the same as claim 25 except that it set forth the claimed invention as an article of manufacture comprising a computer usable medium having readable program code rather than a method and rejected for the same reasons as applied hereinabove.

Referring to claim 39 O'Neal in view of Rizzo teaches the limitations:

(a) "receiving an instruction to insert a new node at an insertion point in said computer-stored hierarchical structure, said nodes of said hierarchical structure stored

as binary encoded values in a computer storage" (O'Neil , Figure 6; Figure 3; Figure 5: Column 8 Lines 36-40, i.e., *Figure 5 and 6 show how data can be inserted (or careted) into a hierarchical data structure, while still maintaining the valuable properties of the position identifier numbering scheme described above*);

(b) "identifying one of, or a combination of the following: a left node ID value closest to the left of said insertion point or a closest right node ID value closest to the right of said insertion point" (O'Neil , Column 8 Lines 58-62, i.e., *In this example, nodes 602 and 604 are assigned position numbers "1.2.1" and "1.2.3", respectively, now becoming sibling nodes to the right of 504 and to the left of 506*);

(c) "calculating a new ID value for node to be inserted" (O'Neal, Column 8 Lines 58-62, i.e., *In this example, nodes 602 and 604 are assigned position numbers "1.2.1" and "1.2.3", respectively, now becoming sibling nodes to the right of 504 and to the left of 506*) "based upon a low key value 0 or a high key value x, said high key value representing a highest binary encodable value and said low key value representing a highest binary encodable value" (Rizzo, Paragraph 0056, i.e., *It is necessary to represent two symbolic* ***values in the key field*** *data range--specifically a representation for positive infinity (+INF) and negative infinity (-INF) is required. It is assumed the key field to be represented with* ***B bits in two's-complement system*** *and to interpret* ***the greatest positive number*** *(2^{B-1}-1) as +INF and* ***the smallest negative number*** *(-2^{B-1}) as -INF. The key value has therefore a range of [$-2^{B-1}+1$, $2^{B-1}-2$], boundary included*), "said calculation performed via one of the following ways: concatenating said left node ID value with one or more high key values and a positive value or concatenating said

left ID value with one or more low key values and a positive value" (O'Neil, Column 8 line 53 through 9 line 3, i.e., *It may be necessary, after tree 500 has been initially constructed, to insert nodes 602 and 604 into tree 500, such that nodes 602 and 604 are child nodes of node 502. Moreover, in the left-to-right ordering among the nodes of tree 500, nodes 602 and 604 may be placed between nodes 504 and 506. In this example, nodes 602 and 604 are assigned position numbers "**1.2.1**" and "**1.2.3**", respectively, now becoming sibling nodes to the right of 504 and to the left of 506. In other words, even number component values are skipped in the initial numbering of the nodes, and are reserved for insertions; the even numbered component values are then ignored in terms of component depth in the tree, becoming siblings of nodes with the same number of odd numbered components. This scheme may be carried out recursively. For example, after nodes 602 and 604 have been inserted into tree 500, it may become necessary to insert node 606 as a further child of node 502 between nodes 602 and 604. Node 606, in this example, receives position number "**1.2.2.1**." This numbering scheme can be carried out for an arbitrary number of insertions, although it may require using arbitrarily long position identifiers. Some insertions on the left or the right of all sibling nodes that are children of a given parent will not require any even numbered components (although insertions to the left of a group of siblings may require a negative odd number--e.g., node 608, which is inserted to the left of the node having position number "1.1", has position number "1.-1".) If node 610 later needs to be inserted in between nodes 608 and 504, the new node 610 will be numbered "1.0.1 (i.e., "0" is the even number between 1 and -1);* 8 Lines 58-62, i.e., *In this example, nodes*

602 and 604 are assigned position numbers "1.2.1" and "1.2.3", respectively, now becoming sibling nodes to the right of 504 and to the left of 506; ***Note that newly inserted node ID "1.2.2.1" of node 606 is derived by concatenating the node ID value of "1.2.1" of node 602 (i.e., dropping '.1" from "1.2.1") and then adding "2" which is "high key" and a positive value which is "1" in this example*)**;** and

(d) "encoding said calculated new ID value and updating said computer storage storing said nodes of said hierarchical structure with said encoded value" (O'Neil , Column 8 Lines 58-62, i.e., *In this example, nodes 602 and 604 are assigned position numbers "1.2.1" and "1.2.3", respectively, now becoming sibling nodes to the right of 504 and to the left of 506*), "wherein order, node ID values, and relationships between parent, child, and siblings in said hierarchical structure of nodes stored in said storage remain unchanged with said insertion of new node" (O'Neil , Column 8 Lines 36-40, i.e., *Figure 5 and 6 show how data can be inserted (or careted) into a hierarchical data structure, while still maintaining the valuable properties of the position identifier numbering scheme described above*).

Claim 40 is rejected on the same basis as claim 25.

Claim 41 is rejected on the same basis as claim 27.

Claim 42 is rejected on the same basis as claim 28.

Claim 43 is rejected on the same basis as claim 29.

Claim 44 is rejected on the same basis as claim 30.

(10) Response to Arguments

Discussion of the Rejection of claims 23-44

Appellant argued that *"Applicants respectfully assert that the combination of O'Neil and Rizzo fail to render obvious many of the features of Applicants' pending claims* " (Applicant's argument, Appeal Brief, page 7 Second Paragraph.

Examiner respectfully disagrees all of the allegations as argued. Examiner, in his previous office action, gave detail explanation of claimed limitation and pointed out exact locations in the cited prior art. Examiner is entitled to give claim limitations their broadest reasonable interpretation in light of the specification. See MPEP 2111 [R-1] Interpretation of Claims-Broadest Reasonable Interpretation.

During patent examination, the pending claims must be 'given the broadest reasonable interpretation consistent with the specification.' Applicant always has the opportunity to amend the claims during prosecution and broad interpretation by the examiner reduces the possibility that the claim, once issued, will be interpreted more broadly than is justified. In re Prater, 162 USPQ 541,550-51 (CCPA 1969).

In response it is pointed out that comparison of the drawings of the claimed invention and the drawing of O'Neal reference is appropriate to highlight the obviousness of O'Neal in view of Rizzo over the claimed invention.

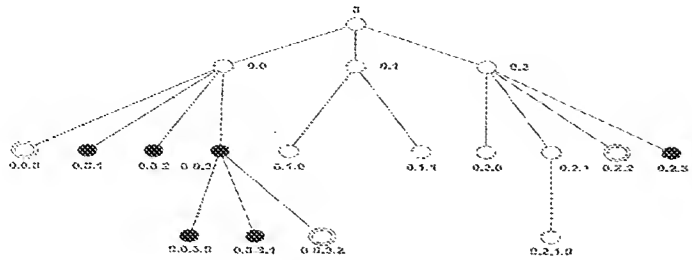

Figure 2b

Prior Art Cited by the Applicant

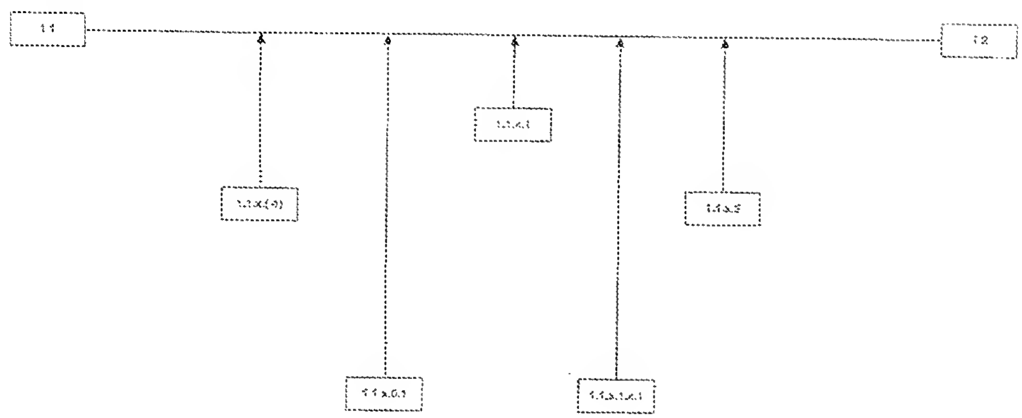

Figure 3

Applicant's Improvement over prior art according to the claimed invention

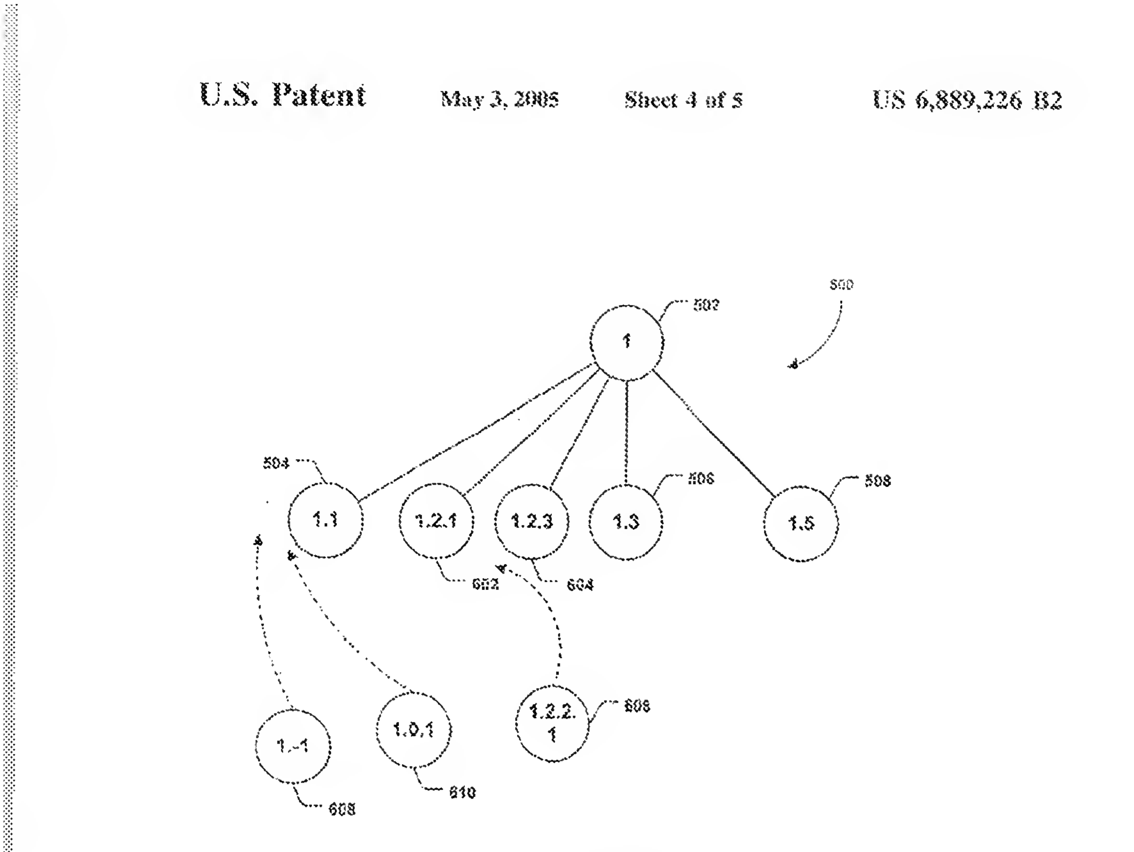

FIG. 6

700

Invention of the O'Neal Patent

In light of the comparison above, it is pointed out that O'Neal in view of Rizzo teaches the claimed features of the application as follows: as per claim 23, O'Neil is directed to a robust computer-based method for updating a computer-stored hierarchical structure of nodes via a node identification technique, said update retaining properties and parent/child relationships of said hierarchical structure without renumbering existing node ID values associated with said hierarchical structure (Abstract, i.e., *The hierarchically organized data is represented as a tree, and each*

node in the tree is assigned a position identifier that represents both depth level of the node within the hierarchy, and its ancestor/descendant relationship to other nodes) and teaches the limitations:

(a) "receiving an instruction to insert a new node at an insertion point in said computer-stored hierarchical structure" (**O'Neil , Figure 6**; Figure 3; Figure 5: Column 8 Lines 36-40, i.e., *Figure 5 and 6 show how data can be inserted (or careted) into a hierarchical data structure, while still maintaining the valuable properties of the position identifier numbering scheme described above*);

(b) "identifying one of, or a combination of the following: a left node ID value closest to the left of said insertion point or a closest right node ID value closest to the right of said insertion point" (O'Neil , Column 8 Lines 58-62, i.e., *In this example, nodes 602 and 604 are assigned position numbers "1.2.1" and "1.2.3", respectively, now becoming sibling nodes to the right of 504 and to the left of 506*; **Particularly note that, in figure of O'Neal, insertion of node 606 identifies the left node ID (i.e., node 602 with ID # 1.2.1) and the right node (i.e., node 604 with ID # 1.2.3) and the ID number of the inserted node (i.e., node 606) has an ID value of 1.2.1.1 which is derived based on the ID values of the left and right nodes (i.e., 1.2.1 and 1.2.3)**);

(c) "calculating a new ID value based upon node ID value(s) identified in (b), said calculated value greater than ID values of nodes to the left of said insertion point and less than ID values of nodes to the right of said insertion point" (O'Neil , Column 8 Lines 58-62, i.e., *In this example, nodes 602 and 604 are assigned position numbers "1.2.1"*

and "1.2.3", respectively, now becoming sibling nodes to the right of 504 and to the left of 506; **Particularly note that, in figure of O'Neal, insertion of node 606 identifies the left node ID (i.e., node 602 with ID # 1.2.1) and the right node (i.e., node 604 with ID # 1.2.3) and the ID number of the inserted node (i.e., node 606) has an ID value of 1.2.1.1 which is derived based on the ID values of the left and right nodes (i.e., 1.2.1 and 1.2.3)**); and

(d) "(encoding) said calculated new ID value and updating said computer storage storing said nodes of said hierarchical structure with said (encoded) value" (O'Neil , Column 8 Lines 58-62, i.e., *In this example, nodes 602 and 604 are* ***assigned position numbers "1.2.1" and "1.2.3"****, respectively, now becoming sibling nodes to the right of 504 and to the left of 506*), "wherein order, node ID values, and relationships between parent, child, and siblings in said hierarchical structure of nodes stored in said storage remain unchanged with said insertion of new node" (O'Neil , Column 8 Lines 36-40, i.e., *Figure 5 and 6 show how data can be inserted (or careted) into a hierarchical data structure,* ***while still "maintaining" the valuable properties of the position identifier*** *numbering scheme described above*).

O'Neil does not explicitly teach the limitation: "encoding" and "said new ID value based upon a low/high key value, said high key value representing a highest encodable value and said low key value representing lowest encodable value".

On the other hand, Rizzo teaches the limitation:

"encoding" (Rizzo, Paragraph 0056, i.e., *the key field to be represented with B bits in two's-complement system*) and "said new ID value based upon a low/high key value, said high key value representing a highest encodable value and said low key value representing lowest encodable value" (Rizzo, Paragraph 0056, i.e., *It is necessary to represent two symbolic* ***values in the key field*** *data range--specifically a representation for positive infinity (+INF) and negative infinity (-INF) is required. It is assumed the key field to be represented with B bits in two's-complement system and to interpret* ***the greatest positive number*** *(2^{B-1}-1) as +INF* ***and the smallest negative number*** *(-2^{B-1}) as -INF. The key value has therefore a range of [-2^{B-1}+1, 2^{B-1}-2], boundary included*).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify the method of O'Neil to add the features of encoding key values into binary numbering system and employing a range of keys/ID's which fall between highest encodable value and lowest encodable value, as taught by Rizzo, to the method of O'Neil so that, in the resultant method, new ID value would be based upon a low/high key value, said high key value representing positive infinity and said low key value representing negative infinity. One would have been motivated to do so in order to have a computer interpret positive infinity as the greatest positive number said computer could process and interpret negative infinity as the smallest number said computer could process (Rizzo, Paragraph 0056) and in order to disable further processing when a data structure is empty or to signal when the data structure is full and overflowing (Rizzo, Paragraph 0055).

As such Examiner affirms that fact that O'Neal in view of Rizzo is obvious over the claim limitations of independent claims 23 and similarly formulated claims of 31 and 39.

Applicant also argued that *"The present claimed invention can be distinguished fro O'Neil because the claimed invention uses the notation of a positive infinity number 'x' (representing the highest encodable value such as 1111) and negative infinity number '0' (representing the lowest encodable value such as 0000)"* (Appellant's argument, page 9 third paragraph).

In response, it is pointed out that O'Neal teaches high key/low key number (as embodied in Figure 6 of O'Neal above, that is, the left node ID (1.2.1, i.e., low key) and the right node ID (1.2.3, i.e., high key). O'Neal does not explicitly teach that said high key is encoded in the highest encodable value (a positive infinity number) and low key is encoded in the lowest encodable value (a negative infinity number). However, Rizzo teaches encoding numerical values in paragraph 0056 as "*the key field to be represented with B bits in two's-complement system*" and teaches representing high key value and low key value by the highest encodable value and the lowest encodable value" (Rizzo, Paragraph 0056, i.e., *It is necessary to represent two symbolic **values in the key field** data range--specifically a representation for positive infinity (+INF) and negative infinity (-INF) is required. It is assumed the key field to be represented with B bits in two's-complement system and to interpret **the greatest positive number** (2^{B-1}-1) as +INF **and the smallest negative number** (-2^{B-1}) as -INF. The key value has therefore a range of [-2^{B-1}+1, 2^{B-1}-2], boundary included*). Therefore, it would have been obvious to a person of ordinary skill in the art to modify the method of O'Neal to add the

features of encode key values/numerical values in binary format and employing the highest encodable value and the lowest encodable value to said high and low keys so that, in the resultant method, node IDs of the left node and right node would be encoded in the highest encodable value and the lowest encodable value.

Appellant argued that "*by contrast, the claimed invention's node insertion/deletion scheme is more robust as it is NOT limited by consideration of even and odd nodes, but is rather dependent on low and high key values*" (Applicant's argument, page 10 last paragraph).

In response, it is pointed out that O'Neal's method inserting node into a branch of a hierarchical structure employs "low and high key values" as embodied in Figure 6 of O'Neal above, that is, the left node ID (1.2.1, i.e., low key) and the right node ID (1.2.3, i.e., high key).

Applicant also argued that "*specifically, if a new node is to be inserted between existing node 4.4 and 4.5, the present invention inserts a new node based on concatenating 'x.1' to the left node 4.4 to form new node '4.4.x.1', wherein 'x' is the high key value. If a new node is to be inserted between 4.1 and the newly created node '4.4.x.1', a low key value of '0' is used to form new node '4.4.x.0.1'*'" (Appellant's argument, page 11 first paragraph).

In response, concatenating left node ID value and right node ID value is taught by O'Neal (O'Neil, Column 8 line 53 through 9 line 3, i.e., *It may be necessary, after tree 500 has been initially constructed, to insert nodes 602 and 604 into tree 500, such that nodes 602 and 604 are child nodes of node 502. Moreover, in the left-to-right ordering among the nodes of tree 500, nodes 602 and 604 may be placed between nodes 504*

*and 506. In this example, nodes 602 and 604 are assigned position numbers "**1.2.1**" and "**1.2.3**", respectively, now becoming sibling nodes to the right of 504 and to the left of 506. In other words, even number component values are skipped in the initial numbering of the nodes, and are reserved for insertions; the even numbered component values are then ignored in terms of component depth in the tree, becoming siblings of nodes with the same number of odd numbered components. This scheme may be carried out recursively. For example, after nodes 602 and 604 have been inserted into tree 500, it may become necessary to insert node 606 as a further child of node 502 between nodes 602 and 604. Node 606, in this example, receives position number "**1.2.2.1.**" This numbering scheme can be carried out for an arbitrary number of insertions, although it may require using arbitrarily long position identifiers. Some insertions on the left or the right of all sibling nodes that are children of a given parent will not require any even numbered components (although insertions to the left of a group of siblings may require a negative odd number--e.g., node 608, which is inserted to the left of the node having position number "1.1", has position number "1.-1".) If node 610 later needs to be inserted in between nodes 608 and 504, the new node 610 will be numbered "1.0.1 (i.e., "0" is the even number between 1 and -1);* 8 Lines 58-62, i.e., *In this example, nodes 602 and 604 are assigned position numbers "1.2.1" and "1.2.3", respectively, now becoming sibling nodes to the right of 504 and to the left of 506;* Note that newly inserted node ID "1.2.2.1" of node 606 is derived by concatenating the node ID value of "1.2.1" of node 602 (i.e., dropping '.1" from "1.2.1") and then adding "2" which is "high key" and a positive value which is "1" in this example).

Appellant argued that *"it should be noted that the secondary references used by the Examiner, i.e., Rizzo, does not relate, in any manner, to the storage, updating or deleting of nodes"* (Appellant's argument, page 11 last paragraph) and that *"applicants are unsure how the Examiner is combining a disparate teaching of a compared value associated with data sorting to Applicant's high and low key values that used in assigning node ID values in a hierarchical structure"* (Appellant's argument, page 12 first paragraph).

In response, it is pointed out that O'Neal and Rizzo are from same fields of sorting data, that is, O'Neal's invention relates to the sorting hierarchically-organized data (O'Neal, Column 1 lines 6-9, i.e., *The present invention relates generally to the field of computing. More particularly, the invention relates to a method of **sorting hierarchically-ordered data***) and Rizzo's invention relates to a data sorter for use in a data processor (Rizzo, Paragraph 0001, i.e., *The present invention is generally directed to processing systems, and, more specifically, to **a data sorter** for use in a data processor).* As such, O'Neal Rizzo are from the same fields of data sorting, and, as a result, it would have been obvious to a person of ordinary skill in the art to combine the features of O'Neal and Rizzo.

Appellant argued that *"Applicants also respectfully assert that the Examiner has failed to show any evidence of why such a mere rang can be interpreted to represent low and high key values that can be combined with the teachings of Rizzo to calculate node values"* (Appellant's argument, page 12 last paragraph) and that *"with respect, the*

Examiner has failed to provide any evidence as to why one of ordinary skill in the art would have been motivated to combine the teachings of O'Neil and Rizzo" (Appellant's argument, page 13 first paragraph).

In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, One would have been motivated to do so in order to have a computer interpret positive infinity as the greatest positive number said computer could process and interpret negative infinity as the smallest number said computer could process (Rizzo, Paragraph 0056) and in order to disable further processing when a data structure is empty or to signal when the data structure is full and overflowing (Rizzo, Paragraph 0055).

Claim 31 is similar to claim 23 except that the claim is directed to an article of manufacture rather than a method and is rejected on the same basis as claim 23. Therefore, applicant is directed to the responses above made regarding claim 23 .

Referring to claim 39 Applicant argued that *"for the Examiner's argument, he/she cites the addition of node 610 prior to the left node 1.1. However, in the Examiner's example, there is no left node ID as the only node ID is the4 right node ID, which is 1.1. With no left node ID, it would have be erroneous to suggest that the addition of node*

610 is in the same as the addition as taught by claim 39" (Appellant's argument, page 17 first paragraph).

In response, it is pointed out that, in the prior office action, Examiner has cited addition of node 606 between nodes 602 and 604 – not node 610, as O'Neal taught in column 8 and column 9 as follows: Column 8 line 53 through 9 line 3, i.e., *It may be necessary, after tree 500 has been initially constructed, to insert nodes 602 and 604 into tree 500, such that nodes 602 and 604 are child nodes of node 502. Moreover, in the left-to-right ordering among the nodes of tree 500, nodes 602 and 604 may be placed between nodes 504 and 506. In this example, nodes 602 and 604 are assigned position numbers "**1.2.1**" and "**1.2.3**", respectively, now becoming sibling nodes to the right of 504 and to the left of 506. In other words, even number component values are skipped in the initial numbering of the nodes, and are reserved for insertions; the even numbered component values are then ignored in terms of component depth in the tree, becoming siblings of nodes with the same number of odd numbered components. This scheme may be carried out recursively. For example, after nodes 602 and 604 have been inserted into tree 500, it may become necessary to insert node 606 as a further child of node 502 between nodes 602 and 604. Node 606, in this example, receives position number "**1.2.2.1**." This numbering scheme can be carried out for an arbitrary number of insertions, although it may require using arbitrarily long position identifiers. Some insertions on the left or the right of all sibling nodes that are children of a given parent will not require any even numbered components (although insertions to the left of a group of siblings may require a negative odd number--e.g., node 608, which is*

inserted to the left of the node having position number "1.1", has position number "1.-1".) If node 610 later needs to be inserted in between nodes 608 and 504, the new node 610 will be numbered "1.0.1 (i.e., "0" is the even number between 1 and -1); 8 Lines 58-62, i.e., *In this example, nodes 602 and 604 are assigned position numbers “1.2.1” and “1.2.3”, respectively, now becoming sibling nodes to the right of 504 and to the left of 506;* **Note that newly inserted node ID “1.2.2.1“ of node 606 is derived by concatenating the node ID value of "1.2.1" of node 602 (i.e., dropping '.1" from "1.2.1") and then adding "2" which is "high key" and a positive value which is “1” in this example.**

In conclusion, it is herewith repeated that claims 23-44 under 35 U.S.C. 103 are unpatentable over the cited prior art.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the Examiner in the Related Appeals and Interferences section of this examiner's answer.

Respectfully Submitted,

/dennis myint/ DM

Dennis Myint
Examiner
AU-2162

September 3, 2008

Conferences:

/Vincent F. Boccio/ VFB

Primary Examiner, Art Unit 2169

Appeal Specialist TC-2100

/John Breene/ JB

Supervisory Patent Examiner, Art Unit 2162